What is claimed is:

1. A method comprising:

generating an interrupt weighted average (IWA) for each of a plurality of processors based on interrupt dispatch information associated with the plurality of processors; and

identifying a target processor from the plurality of processors based on the IWAs to dispatch an interrupt.

2. A method as defined in claim 1, wherein generating the IWA for each of the plurality of processors based on the interrupt dispatch information associated with the plurality of processors comprises generating the IWA for each of the plurality of processors based on at least one of a processor interrupt service age level, a processor interrupt loading history level, and a processor interrupt availability level.

3. A method as defined in claim 1, wherein generating the IWA for each of the plurality of processors based on the interrupt dispatch information associated with the plurality of processors comprises identifying a weight associated with at least one of a processor interrupt service age level, a processor interrupt loading history level, and a processor interrupt availability level.

4. A method as defined in claim 1, wherein generating the IWA for each of the plurality of processors based on the interrupt dispatch information associated with the plurality of processors comprises storing a weight of one or more interrupt load balancing parameters, and calculating the IWA for each of the plurality of processors based the stored weight of the one or more interrupt load balancing parameters.

5. A method as defined in claim 1, wherein identifying the target processor from the plurality of processors based on the IWAs to dispatch the interrupt comprises identifying a processor associated with the highest IWA.

6. A method as defined in claim 1, wherein identifying the target processor from the plurality of processors based on the IWAs to dispatch the interrupt comprises identifying the target processor from the plurality of processors based on the IWAs to dispatch one of a hardware interrupt and a software interrupt.

7. A method as defined in claim 1, further comprising generating an interrupt message to send to the target processor.

8. A machine readable medium storing instructions, which when executed, cause a machine to:

generate an interrupt weighted average (IWA) for each of a plurality of processors based on interrupt dispatch information associated with the plurality of processors; and

identify a target processor from the plurality of processors based on the IWAs to dispatch an interrupt.

9. A machine readable medium as defined in claim 8, wherein the instructions, when executed, cause the machine to generate the IWA for each of the plurality of processors based on the interrupt dispatch information associated with the plurality of processors by generating the IWA for each of the plurality of processors

based on at least one of a processor identifier, a processor interrupt service age level, a processor interrupt loading history level, and a processor interrupt availability level.

10. A machine readable medium as defined in claim 8, wherein the instructions, when executed, cause the machine to generate the IWA for each of the plurality of processors based on the interrupt dispatch information associated with the plurality of processors by identifying a weight associated with at least one of a processor interrupt service age level, a processor interrupt loading history level, and a processor interrupt availability level.

11. A machine readable medium as defined in claim 8, wherein the instructions, when executed, cause the machine to identify the target processor from the plurality of processors based on the IWAs to dispatch the interrupt by identifying a processor associated with the highest IWA.

12. A machine readable medium as defined in claim 8, wherein the instructions, when executed, cause the machine to identify the target processor from the plurality of processors based on the IWAs to dispatch the interrupt by identifying the target processor from the plurality of processors to dispatch one of a hardware interrupt and a software interrupt.

13. A machine readable medium as defined in claim 8, wherein the instructions, which when executed, cause the machine to generate an interrupt message to send to the target processor.

14. A machine readable medium as defined in claim 8, wherein the machine readable medium comprises one of a programmable gate array, application specific integrated circuit, erasable programmable read only memory, read only memory, random access memory, magnetic media, and optical media.

15. An apparatus comprising:

an interrupt load balancing policy register (ILBPR) to store one or more weights corresponding to one or more interrupt load balancing parameters;

a plurality of target processor control registers (TPCRs) to store the interrupt dispatch information associated with a plurality of processors;

a weighted average generator to generate an interrupt weighted average (IWA) for each of the plurality of processors based on the weight corresponding to the one or more interrupt load balancing parameters and the interrupt dispatch information associated with the plurality of processors; and

a target processor selector to identify a target processor from the plurality of processors based on the IWAs to dispatch an interrupt.

16. An apparatus as defined in claim 15, wherein the weight corresponding to one or more interrupt load balancing parameters comprises at least one of a processor interrupt service age weight, a processor interrupt loading history weight, and a processor interrupt availability weight.

17. An apparatus as defined in claim 15, wherein the interrupt dispatch information comprises at least one of a processor identifier, a processor interrupt service

age level, a processor interrupt loading history level, and a processor interrupt availability level.

18. An apparatus as defined in claim 15, wherein the target processor comprises a processor associated with the highest IWA from the plurality of processors.

19. An apparatus as defined in claim 15, wherein the target processor selector generates an interrupt message to send to the target processor.

20. An apparatus as defined in claim 15, wherein the interrupt comprises one of a hardware interrupt and a software interrupt.

21. A processor system comprising:

an input/output controller programmed to request an interrupt; and

a multi-processor programmable interrupt controller (MPIC) programmed to generate an interrupt weighted average (IWA) for each of a plurality of processors based on interrupt dispatch information associated with the plurality of processors, and to identify a target processor from the plurality of processors based on the IWAs to dispatch the interrupt request.

22. A processor system as defined in claim 21, wherein the MPIC is programmed to generate the IWA for each of the plurality of processors based on at least one of a processor identifier, a processor interrupt service age level, a processor interrupt loading history level, and a processor interrupt availability level.

23. A processor system as defined in claim 21, wherein the MPIC is programmed to store weight of the interrupt dispatch information, and to calculate the IWA for each of the plurality of processors based the stored weight of the interrupt dispatch information.

24. A processor system as defined in claim 21, wherein the MPIC is programmed to identify a weight associated with at least one of a processor interrupt service age level, a processor interrupt loading history level, and a processor interrupt availability level corresponding to the plurality of processors.

25. A processor system as defined in claim 21, wherein the MPIC is programmed to identify a processor associated with the highest IWA.

26. A processor system as defined in claim 21, wherein the MPIC is programmed to generate an interrupt message to send to the target processor.

27. A processor system as defined in claim 21, wherein the interrupt comprises one of a hardware interrupt and a software interrupt.

28. A method comprising: /

determining values for a plurality of interrupt load balancing parameters for each of a plurality of processors;

applying a load balancing policy to the values for the plurality of interrupt load balancing parameters to form a plurality of values indicative of an interrupt-related performance of each of the plurality of processors; and identifying one of the plurality of processors as a target processor to receive an interrupt based on the values indicative of the interrupt-related performance of each of the plurality of processors.

29. A method as defined in claim 28, wherein determining values for the plurality of interrupt load balancing parameters for each of the plurality of processors comprises determine values for at least one of a processor interrupt service age parameter, a processor interrupt loading history parameter, and a processor interrupt availability parameter.

30. A method as defined in claim 28, wherein applying the load balancing policy to the values for the plurality of interrupt load balancing parameters to form the plurality of values indicative of the interrupt-related performance of each of the plurality of processors comprises applying an interrupt weighted average to each of the values for at least one of a processor interrupt service age parameter, a processor interrupt loading history parameter, and a processor interrupt availability parameter.